```
ISIS-II PL/M-86 V2.0 COMPILATION OF MODULE ABORT
OBJECT MODULE PLACED IN RABT.OBJ
COMPILER INVOKED BY:  :F0: RABT.PLM XREF OPTIMIZE(3) DEBUG
```

CCP/M-86 2.0 V.2
COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P.O. BOX 579
PACIFIC GROVE, CA 93950
SER # CC-104

```
                $ title ('CCP/M-86 1.0, Abort a Program - RSP')
                $ compact
  1             abort:
                do;

                /* Modified 3/15/83 to force an ATTACH console call */

                $include (:f2:copyrt.lit)
                =
                =  /*
                =    Copyright (C) 1983
                =    Digital Research
                =    P.O. Box 579
                =    Pacific Grove, CA 93950
                =  */
                =
                $include (:f2:comlit.lit)
                =
  2   1         =  declare
                =          lit             literally       'literally',
                =          dcl             lit             'declare',
                =          true            lit             '0ffh',
                =          false           lit             '0',
                =          no              lit             'not',
                =          boolean         lit             'byte',
                =          forever         lit             'while true',
                =          cr              lit             '13',
                =          lf              lit             '10',
                =          tab             lit             '9';
                =
                $include (:f2:mfunc.lit)
                =
                =  /* Concurrent CP/M function numbers */
                =
  3   1         =  dcl          m$prtbuf            lit     '9',
                =                  m$select            lit     '14',
                =                  m$openf             lit     '15',
                =                  m$closef            lit     '16',
                =                  m$deletef           lit     '19',
                =                  m$readf             lit     '20',
                =                  m$writef            lit     '21',
                =                  m$makef             lit     '22',
                =                  m$getlogin          lit     '24',
                =                  m$curdsk            lit     '25',
                =                  m$setdma            lit     '26',
                =                  m$setatt            lit     '30',
                =                  m$setusr            lit     '32',
                =                  m$readrf            lit     '33',
                =                  m$writerf           lit     '34',
                =                  m$resetdrv          lit     '37',
                =                  m$errmode           lit     '45',
```

```
        =               m$dirbios           lit     '50',
        =               m$makeq             lit     '134',
        =               m$openq             lit     '135',
        =               m$deleteq           lit     '136',
        =               m$readq             lit     '137',
        =               m$creadq            lit     '138',
        =               m$writeq            lit     '139',
        =               m$cwriteq           lit     '140',
        =               m$delay             lit     '141',
        =               m$dispatch          lit     '142',
        =               m$setprior          lit     '145',
        =               m$attach            lit     '146',
        =               m$detach            lit     '147',
        =               m$setcns            lit     '148',
        =               m$parse             lit     '152',
        =               m$getcns            lit     '153',
        =               m$sysdat            lit     '154',
        =               m$getpd             lit     '156',
        =               m$abort             lit     '157';
        =
        =   /* Internal calls */
        =
  4  1  =   dcl           mi$sleep            lit     '0212H',
        =               mi$wakeup           lit     '0213H';
        =

            /**** Vax commands for generation:

                $ ccpmsetup                 !Set up environment
                $ asm86 rhabt.a86               !Rsp Header ABorT
                $ plm86 rabt.plm 'p1' 'p2' 'p3' 'p4' optimize(3) debug  !Rsp ABorT
                $ link86 rhabt.obj, rabt.obj to rabt.lnk
                $ loc86 rabt.lnk od(sm(code,dats,data,stack,const))-
                     ad(sm(code(0), dats(10000h)))  ss(stack(0)) to rabt.
                $ h86 rabt

            **** Then, on a micro:
                A>vax rabt.h86 $fans
                A>gencmd rabt data[b1000]
                A>ren abort.rsp=rabt.cmd

            **** Notes:
                The stack is declared in the assemble module, RSPABT.A86.
                The const(ants) come last to force hex generation.
            ****/


  5  1        mon1:
              procedure (func,info) external;
  6  2          declare func byte;
  7  2          declare info address;
  8  2        end mon1;

  9  1        mon2:
              procedure (func,info) byte external;
 10  2          declare func byte;
 11  2          declare info address;
```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
12  2           end mon2;

13  1         mon3:
                procedure (f,a) address external;
14  2             dcl f byte, a address;
15  2           end mon3;

16  1       patch: procedure public;    /* dummy area for patching code segments */
17  2         declare i address;
              /* first statement is 9 bytes, rest are 5 bytes */
18  2           i=i+5;  i=i+5;  i=i+5;  i=i+5;  i=i+5;
23  2           i=i+5;  i=i+5;  i=i+5;  i=i+5;  i=i+5;     /* about 54 bytes */
28  2         end patch;

          $include (:f2:proces.lit)
        =
        =   /*
        =       Proces Literals MP/M-8086 II
        =   */
        =
29  1   =   declare pnamsiz literally '8';
        =
30  1   =   declare pd$hdr literally 'structure
        =     (link word,thread word,stat byte,prior byte,flag word,
        =     name (8) byte,uda word,dsk byte,user byte,ldsk byte,luser byte,
        =     mem word';
        =
31  1   =   declare pd$structure literally 'pd$hdr,
        =     dvract word,wait word,org byte,net byte,parent word,
        =     cns byte,abort byte,conmode word,lst byte,sf3 byte,sf4 byte,sf5 byte,
        =     reservd (4) byte,pret word,scratch word)';
        =
32  1   =     declare psrun                 lit '00',
        =             pspoll                lit '01',
        =             psdelay               lit '02',
        =             psswap                lit '03',
        =             psterm                lit '04',
        =             pssleep               lit '05',
        =             psdq                  lit '06',
        =             psnq                  lit '07',
        =             psflagwait            lit '08',
        =             psciowait             lit '09';
        =
33  1   =     declare pf$sys                lit '00001h',
        =             pf$keep               lit '00002h',
        =             pf$kernal             lit '00004h',
        =             pf$pure               lit '00008h',
        =             pf$table              lit '00010h',
        =             pf$resource           lit '00020h',
        =             pf$raw                lit '00040h',
        =             pf$ctlc               lit '00080h',
        =             pf$active             lit '00100h',
        =             pf$tempkeep           lit '00200h',
        =             pf$ctld               lit '00400h',
        =             pf$childabort         lit '00800h',
        =             pf$noctls             lit '01000h';
        =
```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
34   1  =     declare pcm$ll                lit '00001h',
        =             pcm$ctls              lit '00002h',
        =             pcm$rout              lit '00004h',
        =             pcm$ctlc              lit '00008h',
        =             pcm$ctlo              lit '00080h',
        =             pcm$rsx               lit '00300h';

35   1        dcl pd$pointer pointer;
36   1        dcl pd$ptr structure (offset word, segment word) at(@pd$pointer);
37   1        dcl pd based pd$pointer pd$structure;

38   1        dcl rsplink word external;

            $include (:f2:qd.lit)
        =
        =   /* Queue Descriptor */
        =
39   1  =   dcl qnamsiz lit '8';
        =
40   1  =   dcl qd$structure lit 'structure(
        =     link  word,
        =     net byte,
        =     org byte,
        =     flags word,
        =     name(qnamsiz) byte,
        =     msglen word,
        =     nmsgs word,
        =     dq word,
        =     nq word,
        =     msgcnt word,
        =     msgout word,
        =     buffer word)';
        =
        =   /* queue flag values */
        =
41   1  =   dcl qf$mx       lit '001h'; /* Mutual Exclusion */
42   1  =   dcl qf$keep     lit '002h'; /* NO DELETE        */
43   1  =   dcl qf$hide     lit '004h'; /* Not User writable    */
44   1  =   dcl qf$rsp      lit '008h'; /* rsp queue        */
45   1  =   dcl qf$table    lit '010h'; /* from qd table    */
46   1  =   dcl qf$rpl      lit '020h'; /* rpl queue        */
47   1  =   dcl qf$dev      lit '040h'; /* device queue     */
        =
        =   /* Queue Parameter Block */
        =
48   1  =   dcl qpb$structure lit 'structure(
        =     flgs    byte,
        =     net     byte,
        =     qaddr   word,
        =     nmsgs   word,
        =     buffptr word,
        =     name (qnamsiz) byte )';
        =

49   1        dcl abt$qd$buf (131) byte;
50   1        dcl abt$cmd structure(
               pd address, tail (129) byte);
```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
51   1        dcl abt$qpb qpb$structure initial(0,0,0,1,.abt$cmd,'ABORT   ');
52   1        dcl abt$qd qd$structure initial (
               0,0,0,qf$keep + qf$rsp,'ABORT   ',131,1,0,0,0,0,.abt$qd$buf);

53   1        dcl fcb (32) byte;
54   1        dcl pfcb structure (
               filename address,
               fcbadr address) initial (.abt$cmd.tail, .fcb);

55   1        declare abort$pb structure (
               pd address,
               term address,
               cns byte,
               net byte,
               pname (8) byte) initial (
               0,00ffh,0,0,'        ');

56   1        dcl i byte;
57   1        dcl console word;
58   1        dcl mpm$86 lit '1130h';

               /*
                 Main Program
               */

59   1      plm$start:
               procedure public;

60   2          call mon1(m$makeq,.abt$qd);          /* make ABORT queue      */
61   2          call mon1(m$openq,.abt$qpb);         /* open it               */
62   2          pd$ptr.segment = rsplink;
63   2          call mon1(m$setprior,200);           /* back to the same as transients */

64   2          do while true;
65   3            call mon1(m$readq,.abt$qpb);
66   3            pd$ptr.offset = abt$cmd.pd;        /* set console to same       */
67   3            call mon1(m$setcns, pd.cns);  /* of who typed ABORT        */
68   3            abort$pb.cns = pd.cns;
69   3            pfcb.filename = mon3(m$parse, .pfcb);
                                                    /* get name of program to abort */
70   3            call move (8,.fcb(1),.abort$pb.pname);
               /*       fcb(9)='$';   DEBUG
                     call mon1(m$prtbuf, .fcb(1)); */
71   3            if pfcb.filename <> 0 then        /* console number specified     */
72   3            do;
73   4              pfcb.filename = mon3(m$parse, .pfcb);
74   4              i = 1; console = 0;
76   4              do while fcb(i) <> ' ' and i < 4;
77   5                if (fcb(i) := fcb(i) - '0') <= 9 then
78   5                do;
79   6                  console = fcb(i) + 10 * console;
80   6                  i = i + 1;
81   6                end;
                       else
82   5                  i = 255;                      /* non - numeric */
83   5                end;
84   4              if console > 253 or i = 255 then
```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
 85   4            do;
 86   5              call mon1(m$prtbuf, .(cr,lf, 'Illegal Console, Use 0-253 $'));
 87   5              abort$pb.cns = 0ffh;
 88   5            end;
                    else
 89   4              abort$pb.cns = low(console);
 90   4          end;

 91   3          if abort$pb.cns <> 0ffh then
 92   3            if mon2(m$abort, .abort$pb) = 0ffh then
 93   3            do;
 94   4              call mon1(m$prtbuf, .(cr,lf, 'Abort Failed.','$'));
 95   4            end;
                              /* abort first PD found with same name and console */
                              /* consistent with MP/M-80 II but not MP/M 1.x     */
 96   3          call mon1(m$attach,0);    /* make sure we own the console before */
 97   3          call mon1(m$detach, 0);   /* calling detach                      */
 98   3          pfcb.filename = .abt$cmd.tail;
 99   3        end;
100   2      end plmstart;
101   1      end abort;
```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

CROSS-REFERENCE LISTING

DEFN	ADDR	SIZE	NAME, ATTRIBUTES, AND REFERENCES											
13	0000H	2	A.	WORD PARAMETER	14									
1	0002H		ABORT.	PROCEDURE STACK=0000H										
37	0021H	1	ABORT.	BYTE MEMBER(PD)										
55	0036H	14	ABORTPB.	STRUCTURE INITIAL		68	70	87	89	91	92			
50	00C9H	131	ABTCMD	STRUCTURE	51	54	66	98						
52	0016H	28	ABTQD.	STRUCTURE INITIAL			60							
49	0046H	131	ABTQDBUF	BYTE ARRAY(131)		52								
51	0006H	16	ABTQPB	STRUCTURE INITIAL			61	65						
2			BOOLEAN.	LITERALLY										
52	001AH	2	BUFFER	WORD MEMBER(ABTQD)										
51	0006H	2	BUFFPTR.	WORD MEMBER(ABTQPB)										
37	0020H	1	CNS.	BYTE MEMBER(PD)		67	68							
55	0004H	1	CNS.	BYTE MEMBER(ABORTPB)			68	87	89	91				
37	0022H	2	CONMODE.	WORD MEMBER(PD)										
57	0044H	2	CONSOLE.	WORD	75	79	84	89						
2			CR	LITERALLY		86	94							
2			DCL.	LITERALLY										
54	0012H	2	DQ	WORD MEMBER(ABTQD)										
37	0012H	1	DSK.	BYTE MEMBER(PD)										
37	0018H	2	DVRACT	WORD MEMBER(PD)										
13	0000H	1	F.	BYTE PARAMETER		14								
2			FALSE.	LITERALLY										
53	014CH	32	FCB.	BYTE ARRAY(32)		54	70	76	77	79				
54	0002H	2	FCBADR	WORD MEMBER(PFCB)										
54	0000H	2	FILENAME	WORD MEMBER(PFCB)			69	71	73	98				
37	0006H	2	FLAG	WORD MEMBER(PD)										
52	0004H	2	FLAGS.	WORD MEMBER(ABTQD)										
51	0000H	1	FLGS	BYTE MEMBER(ABTQPB)										
2			FOREVER.	LITERALLY										
5	0000H	1	FUNC	BYTE PARAMETER		6								
9	0000H	1	FUNC	BYTE PARAMETER		10								
17	0000H	2	I.	WORD	18	19	20	21	22	23	24	25	26	27
56	016CH	1	I.	BYTE	74	76	77	79	80	82	84			
5	0000H	2	INFO	WORD PARAMETER		7								
9	0000H	2	INFO	WORD PARAMETER		11								
37	0014H	1	LDSK	BYTE MEMBER(PD)										
2			LF	LITERALLY		86	94							
37	0000H	2	LINK	WORD MEMBER(PD)										
52	0000H	2	LINK	WORD MEMBER(ABTQD)										
2			LIT.	LITERALLY	2	3	4	32	33	34	39	40	41	42
				44	45	46	47	48	58					
			LOW.	BUILTIN		89								
37	0024H	1	LST.	BYTE MEMBER(PD)										
37	0015H	1	LUSER.	BYTE MEMBER(PD)										
3			MABORT	LITERALLY		92								
3			MATTACH.	LITERALLY		96								
3			MCLOSEF.	LITERALLY										
3			MCREADQ.	LITERALLY										
3			MCURDSK.	LITERALLY										

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
  3                 MCWRITEQ . . . . .   LITERALLY
  3                 MDELAY . . . . . .   LITERALLY
  3                 MDELETEF . . . . .   LITERALLY
  3                 MDELETEQ . . . . .   LITERALLY
  3                 MDETACH. . . . . .   LITERALLY          97
  3                 MDIRBIOS . . . . .   LITERALLY
  3                 MDISPATCH. . . . .   LITERALLY
 37  0016H      2  MEM. . . . . . . .   WORD MEMBER(PD)
  3                 MERRMODE . . . . .   LITERALLY
  3                 MGETCNS. . . . . .   LITERALLY
  3                 MGETLOGIN. . . . .   LITERALLY
  3                 MGETPD . . . . . .   LITERALLY
  4                 MISLEEP. . . . . .   LITERALLY
  4                 MIWAKEUP . . . . .   LITERALLY
  3                 MMAKEF . . . . . .   LITERALLY
  3                 MMAKEQ . . . . . .   LITERALLY          60
  5  0000H          MON1 . . . . . . .   PROCEDURE EXTERNAL(0) STACK=0000H          60   61   63   65   67   85
                                            94   96   97
  9  0000H          MON2 . . . . . . .   PROCEDURE BYTE EXTERNAL(1) STACK=0000H          92
 13  0000H          MON3 . . . . . . .   PROCEDURE WORD EXTERNAL(2) STACK=0000H          69   73
  3                 MOPENF . . . . . .   LITERALLY
  3                 MOPENQ . . . . . .   LITERALLY          61
                    MOVE . . . . . . .   BUILTIN            70
  3                 MPARSE . . . . . .   LITERALLY          69   73
 58                 MPM86. . . . . . .   LITERALLY
  3                 MPRTBUF. . . . . .   LITERALLY          86   94
  3                 MREADF . . . . . .   LITERALLY
  3                 MREADQ . . . . . .   LITERALLY          65
  3                 MREADRF. . . . . .   LITERALLY
  3                 MRESETDRV. . . . .   LITERALLY
  3                 MSELECT. . . . . .   LITERALLY
  3                 MSETATT. . . . . .   LITERALLY
  3                 MSETCNS. . . . . .   LITERALLY          67
  3                 MSETDMA. . . . . .   LITERALLY
  3                 MSETPRIOR. . . . .   LITERALLY          63
  3                 MSETUSR. . . . . .   LITERALLY
 52  0016H      2  MSGCNT . . . . . .   WORD MEMBER(ABTQD)
 52  000EH      2  MSGLEN . . . . . .   WORD MEMBER(ABTQD)
 52  0018H      2  MSGOUT . . . . . .   WORD MEMBER(ABTQD)
  3                 MSYSDAT. . . . . .   LITERALLY
  3                 MWRITEF. . . . . .   LITERALLY
  3                 MWRITEQ. . . . . .   LITERALLY
  3                 MWRITERF . . . . .   LITERALLY
 52  0006H      8  NAME . . . . . . .   BYTE ARRAY(8) MEMBER(ABTQD)
 51  0008H      8  NAME . . . . . . .   BYTE ARRAY(8) MEMBER(ABTQPB)
 37  0008H      8  NAME . . . . . . .   BYTE ARRAY(8) MEMBER(PD)
 55  0005H      1  NET. . . . . . . .   BYTE MEMBER(ABORTPB)
 52  0002H      1  NET. . . . . . . .   BYTE MEMBER(ABTQD)
 51  0001H      1  NET. . . . . . . .   BYTE MEMBER(ABTQPB)
 37  001DH      1  NET. . . . . . . .   BYTE MEMBER(PD)
 52  0010H      2  NMSGS. . . . . . .   WORD MEMBER(ABTQD)
 51  0004H      2  NMSGS. . . . . . .   WORD MEMBER(ABTQPB)
  2                 NO . . . . . . . .   LITERALLY
 52  0014H      2  NQ . . . . . . . .   WORD MEMBER(ABTQD)
 36  0000H      2  OFFSET . . . . . .   WORD MEMBER(PDPTR)          66
 52  0003H      1  ORG. . . . . . . .   BYTE MEMBER(ABTQD)
 37  001CH      1  ORG. . . . . . . .   BYTE MEMBER(PD)
```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
37  001EH     2  PARENT . . . . . .   WORD MEMBER(PD)
16  0002H    60  PATCh. . . . . .     PROCEDURE PUBLIC STACK=0002H
34             PCM11. . . . . .       LITERALLY
34             PCMCTLC. . . . . .     LITERALLY
34             PCMCTLD. . . . . .     LITERALLY
34             PCMCTLS. . . . . .     LITERALLY
34             PCMROUT. . . . . .     LITERALLY
34             PCMRSX . . . . . .     LITERALLY
55  0000H     2  PD . . . . . .       WORD MEMBER(ABORTPB)
50  0000H     2  PD . . . . . .       WORD MEMBER(ABTCMD)          66
37  0000H    48  PD . . . . . .       STRUCTURE BASED(PDPOINTER)               67   68
30             PDHDR. . . . . .       LITERALLY          37
35  0002H     4  PDPOINTER. . . . .   POINTER            36    37   67   68
36  0002H     4  PDPTR. . . . . .     STRUCTURE AT             62   66
31             PDSTRUCTURE. . . .     LITERALLY          37
33             PFACTIVE . . . . .     LITERALLY
54  0032H     4  PFCB . . . . . .     STRUCTURE INITIAL            69   71   73   98
33             PFCHILDABORT . . .     LITERALLY
33             PFCTLC . . . . . .     LITERALLY
33             PFCTLD . . . . . .     LITERALLY
33             PFKEEP . . . . . .     LITERALLY
33             PFKERNAL . . . . .     LITERALLY
33             PFNOCTLS . . . . .     LITERALLY
33             PFPURE . . . . . .     LITERALLY
33             PFRAW. . . . . .       LITERALLY
33             PFRESOURCE . . . .     LITERALLY
33             PFSYS. . . . . .       LITERALLY
33             PFTABLE. . . . . .     LITERALLY
33             PFTEMPKEEP . . . .     LITERALLY
59  003EH   322  PLMSTART . . . . .   PROCEDURE PUBLIC STACK=0008H
55  0006H     8  PNAME. . . . . .     BYTE ARRAY(8) MEMBER(ABORTPB)             70
29             PNAMSIZ. . . . . .     LITERALLY
37  002CH     2  PRET . . . . . .     WORD MEMBER(PD)
37  0005H     1  PRIOR. . . . . .     BYTE MEMBER(PD)
32             PSCIOWAIT. . . . .     LITERALLY
32             PSDELAY. . . . . .     LITERALLY
32             PSDQ . . . . . .       LITERALLY
32             PSFLAGWAIT . . . .     LITERALLY
32             PSNQ . . . . . .       LITERALLY
32             PSPOLL . . . . . .     LITERALLY
32             PSRUN. . . . . .       LITERALLY
32             PSSLEEP. . . . . .     LITERALLY
32             PSSWAP . . . . . .     LITERALLY
32             PSTERM . . . . . .     LITERALLY
51  0002H     2  QADDR. . . . . .     WORD MEMBER(ABTQPB)
40             QDSTRUCTURE. . . .     LITERALLY          52
47             QFDEV. . . . . .       LITERALLY
43             QFHIDE . . . . . .     LITERALLY
42             QFKEEP . . . . . .     LITERALLY          52
41             QFMX . . . . . .       LITERALLY
46             QFRPL. . . . . .       LITERALLY
44             QFRSP. . . . . .       LITERALLY          52
45             QFTABLE. . . . . .     LITERALLY
39             QNAMSIZ. . . . . .     LITERALLY          51   52
48             QPBSTRUCTURE . . .     LITERALLY          51
37  0028H     4  RESERVD. . . . . .   BYTE ARRAY(4) MEMBER(PD)
38  0000H     2  RSPLINK. . . . . .   WORD EXTERNAL(3)             62
```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
 37  002EH     2  SCRATCH. . . . . .   WORD MEMBER(PD)
 36  0002H     2  SEGMENT. . . . . .   WORD MEMBER(PDPTR)          62
 37  0025H     1  SF3. . . . . .       BYTE MEMBER(PD)
 37  0026H     1  SF4. . . . . .       BYTE MEMBER(PD)
 37  0027H     1  SF5. . . . . .       BYTE MEMBER(PD)
 37  0004H     1  STAT . . . . . .     BYTE MEMBER(PD)
  2               TAB. . . . . .       LITERALLY
 50  0002H   129  TAIL . . . . . .     BYTE ARRAY(129) MEMBER(ABTCMD)      54   98
 55  0002H     2  TERM . . . . . .     WORD MEMBER(ABORTPB)
 37  0002H     2  THREAD . . . . . .   WORD MEMBER(PD)
  2               TRUE . . . . . .     LITERALLY        64
 37  0010H     2  UDA. . . . . .       WORD MEMBER(PD)
 37  0013H     1  USER . . . . . .     BYTE MEMBER(PD)
 37  0014H     2  WAIT . . . . . .     WORD MEMBER(PD)


MODULE INFORMATION:

     CODE AREA SIZE     = 0180H    384D
     CONSTANT AREA SIZE = 002EH     46D
     VARIABLE AREA SIZE = 0160H    365D
     MAXIMUM STACK SIZE = 0008H      8D
     300 LINES READ
     0 PROGRAM ERROR(S)

END OF PL/M-86 COMPILATION
```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P.O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
RABT.MP2:1                    27-APR-1983 09:31                Page 1

ISIS-II MCS-86 LOCATER, V1.2 INVOKED BY:
:F0: RABT.LNK OD(SM(CODE,DATA,DATA,STACK,CONST)) AD(SM(CODE(0), DATA(10000H))) SS(STACK(0)) TO RABT.
WARNING 26:  DECREASING SIZE OF SEGMENT
  SEGMENT:  STACK

SYMBOL TABLE OF MODULE RABT
READ FROM FILE RABT.LNK
WRITTEN TO FILE RABT.

BASE    OFFSET TYPE SYMBOL                BASE    OFFSET TYPE SYMBOL

1000H   0000H  PUB  BSPLINK               0000H   0072H  PUB  PLMSTART
0000H   0036H  PUB  PATCH                 0000H   0025H  PUB  XDOS
0000H   0025H  PUB  MON1                  0000H   0025H  PUB  MON2
0000H   0025H  PUB  MON3                  0000H   0025H  PUB  MON4

ABORT:  SYMBOLS AND LINES
1000H   02E0H  SYM  MEMORY                0000H   0036H  SYM  PATCH
1000H   0140H  SYM  I                     1000H   0142H  SYM  PDPOINTER
1000H   0142H  SYM  PDPTR                 1000H   0142H  BAS  PD
1000H   0186H  SYM  ABTQDBUF              1000H   0209H  SYM  ABTCMD
1000H   0146H  SYM  ABTQPB                1000H   0156H  SYM  ABTQD
1000H   028CH  SYM  FCB                   1000H   0172H  SYM  PFCB
1000H   0176H  SYM  ABORTPB               1000H   02ACH  SYM  T
1000H   0184H  SYM  CONSOLE               0000H   0072H  SYM  PLMSTART
0000H   0036H  LIN     16                 0000H   0039H  LIN     18
0000H   0044H  LIN     19                 0000H   0049H  LIN     20
0000H   004EH  LIN     21                 0000H   0053H  LIN     22
0000H   0058H  LIN     23                 0000H   005DH  LIN     24
0000H   0062H  LIN     25                 0000H   0067H  LIN     26
0000H   006CH  LIN     27                 0000H   0070H  LIN     28
0000H   0072H  LIN     59                 0000H   0075H  LIN     60
0000H   007FH  LIN     61                 0000H   0089H  LIN     62
0000H   008FH  LIN     63                 0000H   0099H  LIN     64
0000H   0099H  LIN     65                 0000H   00A3H  LIN     66
0000H   00A9H  LIN     67                 0000H   00BAH  LIN     68
0000H   00C5H  LIN     69                 0000H   00D2H  LIN     70
0000H   00E0H  LIN     71                 0000H   00EAH  LIN     73
0000H   00F7H  LIN     74                 0000H   00FCH  LIN     75
0000H   0102H  LIN     76                 0000H   0122H  LIN     77
0000H   0130H  LIN     79                 0000H   0143H  LIN     80
0000H   0147H  LIN     81                 0000H   0149H  LIN     82
0000H   014EH  LIN     83                 0000H   0150H  LIN     84
0000H   015FH  LIN     86                 0000H   0169H  LIN     87
0000H   016CH  LIN     88                 0000H   0170H  LIN     89
0000H   0176H  LIN     91                 0000H   017DH  LIN     92
0000H   0186H  LIN     94                 0000H   0195H  LIN     96
0000H   019FH  LIN     97                 0000H   01A9H  LIN     98
0000H   01AFH  LIN     99                 0000H   0132H  LIN    100
0000H   0036H  LIN    101

MEMORY MAP OF MODULE RABT
READ FROM FILE RABT.LNK
WRITTEN TO FILE RABT.

SEGMENT MAP

START     STOP      LENGTH ALIGN NAME              CLASS
```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

00000H	00153H	0154H	G	CODE	CODE
10000H	1013FH	0140H	G	DATS	DATA
10140H	102ACH	016DH	W	DATA	DATA
102ACH	102ADH	0000H	W	STACK	STACK
102ADH	102D6H	002EH	W	CONST	CONST
102E0H	102E0H	0000H	G	??SEG	
102E0H	102E0H	0000H	W	MEMORY	MEMORY

GROUP MAP

ADDRESS	GROUP OR SEGMENT NAME
10000H	DGROUP
	DATS
	CONST
	DATA
00000H	CGROUP
	CODE

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

ISIS-II MCS-86 MACRO ASSEMBLER V2.1 ASSEMBLY OF MODULE RHABT
OBJECT MODULE PLACED IN :F0:RHABT.OPJ
ASSEMBLER INVOKED BY: :F0: RHABT.A86

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
LOC  OBJ                  LINE     SOURCE

                             1     ;       Code and Data Interface for ABORT.RSP
                             2     ;       August 10, 1981
                             3     ;       July   27, 1982      (updated)
                             4     ;       Jan 1983  whf - added reg save to xdos
                             5     ;       Mar 1983  dh  - shrunk to resonable size
                             6
                             7     name    rhabt           ;Rsp Header AborT
                             8
                             9     cgroup  group   code
                            10     dgroup  group   dats
                            11             public  xdos,mon1,mon2,mon3,mon4
                            12             public rsplink
                            13             extrn  plmstart:near
                            14
                            15     assume   cs:cgroup,ds:dgroup
                            16
----                        17     dats    segment "DATA"
0000                        18     org     0
  0010                      19     rsphdr_len      equ     16
  0030                      20     pd_len          equ     30H
  0100                      21     uda_len         equ     100H
  0000                      22     rsp_top         equ     0
  0010                      23     rsp_pd          equ     rsp_top + rsphdr_len
  0040                      24     rsp_uda         equ     rsp_pd + pd_len
  0140                      25     rsp_bottom      equ     rsp_uda + uda_len
0000                        26             org     rsp_top
                            27                                     ;RSP header
0000 0000                   28     rsplink dw      0               ;becomes system data page paragraph
0002 0000                   29     sdatvar dw      0
0004 00                     30     ncopies db      0
0005 0000                   31             dw      0,0,0,0, 0
0007 0000
0009 0000
000B 0000
000D 0000
000F 00                     32             db      0
0010                        33             org     rsp_pd
0010 0000                   34     pd      dw      0,0             ;link fields
0012 0000
0014 00                     35             db      0               ;status
0015 BE                     36             db      190             ;priority
0016 0300                   37             dw      3               ;flags - system and keep
0018 41424F52542020         38             db      'ABORT   '      ;name
     20
0020 0400                   39             dw      rsp_uda/10h     ;uda paragraph
0022 00                     40             db      0,0             ;disk,user
0023 00
0024 00                     41             db      0,0             ;ldisk,luser
0025 00
0026 0000                   42             dw      0               ;puremem - not re-entrant
```

```
LOC  OBJ                  LINE     SOURCE

                          43                                   ;rest of pd
                          44
0040                      45       org      rsp_uda             ;start of uda
0040 0000                 46       uda      dw       0
0042 0000                 47                dw       0          ;DMA must be explicitly set
0044 0000                 48                dw       0,0,0,0, 0,0,0,0, 0,0,0,0
0046 0000
0048 0000
004A 0000
004C 0000
004E 0000
0050 0000
0052 0000
0054 0000
0056 0000
0058 0000
005A 0000
005C 0000                 49                dw       0,0,0,0, 0,0,0,0, 0,0,0,0
005E 0000
0060 0000
0062 0000
0064 0000
0066 0000
0068 0000
006A 0000
006C 0000
006E 0000
0070 0000
0072 0000
0074 3A01                 50                dw       offset stk_top
00A0                      51       org      rsp_uda + 60H
00A0 01                   52                db       1                   ;insys = 1, always use UDA stack
00A1 00                   53                db       0
00A2 CCCC                 54                dw       0cccch,0cccch,0cccch,0cccch      ;62
00A4 CCCC
00A6 CCCC
00A8 CCCC
00AA CCCC                 55                dw       0cccch,0cccch,0cccch,0cccch      ;6A
00AC CCCC
00AE CCCC
00B0 CCCC
00B2 CCCC                 56                dw       0cccch,0cccch,0cccch,0cccch      ;72
00B4 CCCC
00B6 CCCC
00B8 CCCC
00BA CCCC                 57                dw       0cccch,0cccch,0cccch,0cccch      ;7A
00BC CCCC
00BE CCCC
00C0 CCCC
00C2 CCCC                 58                dw       0cccch,0cccch,0cccch,0cccch      ;82
00C4 CCCC
00C6 CCCC
00C8 CCCC
00CA CCCC                 59                dw       0cccch,0cccch,0cccch,0cccch      ;8A
00CC CCCC
```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
LOC  OBJ                  LINE     SOURCE

00CE CCCC
00D0 CCCC
0002 CCCC                   60             dw      0cccch,0cccch,0cccch,0cccch     ;92
00D4 CCCC
00D6 CCCC
00D8 CCCC
00DA CCCC                   61             dw      0cccch,0cccch,0cccch,0cccch     ;9A
00DC CCCC
00DE CCCC
00E0 CCCC
00E2 CCCC                   62             dw      0cccch,0cccch,0cccch,0cccch     ;A2
00E4 CCCC
00E6 CCCC
00E8 CCCC
00EA CCCC                   63             dw      0cccch,0cccch,0cccch,0cccch     ;AA
00EC CCCC
00EE CCCC
00F0 CCCC
00F2 CCCC                   64             dw      0cccch,0cccch,0cccch,0cccch     ;B2
00F4 CCCC
00F6 CCCC
00F8 CCCC
00FA CCCC                   65             dw      0cccch,0cccch,0cccch,0cccch     ;BA
00FC CCCC
00FE CCCC
0100 CCCC
0102 CCCC                   66             dw      0cccch,0cccch,0cccch,0cccch     ;C2
0104 CCCC
0106 CCCC
0108 CCCC
010A CCCC                   67             dw      0cccch,0cccch,0cccch,0cccch     ;CA
010C CCCC
010E CCCC
0110 CCCC
0112 CCCC                   68             dw      0cccch,0cccch,0cccch,0cccch     ;D2
0114 CCCC
0116 CCCC
0118 CCCC
011A CCCC                   69             dw      0cccch,0cccch,0cccch,0cccch     ;DA
011C CCCC
011E CCCC
0120 CCCC
0122 CCCC                   70             dw      0cccch,0cccch,0cccch,0cccch     ;E2
0124 CCCC
0126 CCCC
0128 CCCC
012A CCCC                   71             dw      0cccch,0cccch,0cccch,0cccch     ;EA
012C CCCC
012E CCCC
0130 CCCC
0132 CCCC                   72             dw      0cccch,0cccch,0cccch,0cccch     ;F2
0134 CCCC
0136 CCCC
0138 CCCC
013A 0000            E      73     stk_top dw      plmstart,0,0                    ;initial IRET
```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
LOC  OBJ                  LINE     SOURCE

013C 0000
013E 0000
                           74
0140                       75     org     rsp_bottom
                           76
  0140                     77     datsend equ     offset $
----                       78     dats ends
                           79
----                       80     code    segment public 'CODE'
0000                       81             org     0
                           82
0000 434F5059524947        83             db      'COPYRIGHT (C) 1982,'
     48542028432920
     313938322C
0013 20444947495441        84             db      ' DIGITAL RESEARCH '
     4C205245534541
     52434820
                           85
0025                       86     xdos    proc
0025 55                    87             push    bp
0026 8BEC                  88             mov     bp,sp
0028 8B5604                89             mov     dx,[bp+4]
002B 8B4E06                90             mov     cx,[bp+6]
002E CDE0                  91             int     224
0030 5D                    92             pop     bp
0031 C20400                93             ret     4
                           94     xdos    endp
                           95
  0025                     96     mon1    equ     xdos
  0025                     97     mon2    equ     xdos
  0025                     98     mon3    equ     xdos
  0025                     99     mon4    equ     xdos
----                      100     code    ends
                          101     end

ASSEMBLY COMPLETE, NO ERRORS FOUND
```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____